グレブナ基底を用いた最尤複号アルゴリズムについて (グレブナ-基底の理論的有効性と実践的有効性)

AUTHOR(S):

池上, 大介

CITATION:

池上, 大介. グレブナ基底を用いた最尤複号アルゴリズムについて (グレブナ-基底の理論的有効性と実践的有効性). 数理解析研究所講究録 2002, 1289: 110-121

ISSUE DATE:

2002-09

URL:

http://hdl.handle.net/2433/42509

数理解析研究所講究録 1289 巻 2002 年 110-121

グレブナ基底を用いた最尤復号アルゴリズムについて

奈良先端科学技術大学院大学・情報科学研究科 池上 大介 (IKEGAMI Daisuke) [1]
Graduate School of Information Science,
Nara Institute of Science and Technology

1 はじめに

符号理論は 1940 年代に Golay, Hamming, Shannon らによって始められた情報理論の一派である。その後、符号理論は数学の発展とともに進歩し続け、多項式環論や代数幾何学などと融合して、魅力的な研究分野の一つとなっている。現在、符号理論と呼ばれる理論は主に次の二つである:

- 誤り訂正符号理論
- 情報源符号化理論

本原稿は、誤り訂正符号理論における問題の一つ「最速な最尤復号アルゴリズムの構成」を対象とする。一方、後者の情報源符号化理論は情報の圧縮・伸長を考察するものであり、本稿では取り扱わない。

誤り訂正符号理論とは、"雑音のある通信路" において通信を行うための技術を考察する。送信者甲が受信者乙に雑音のある通信路を通じて情報を送信することを考える (図 1)。

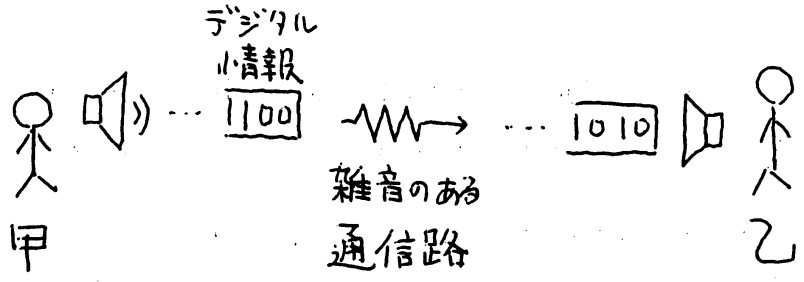

図 1: 雑音のある通信路

確率的に定義される雑音に対して、受信者が送信語を推定する操作を復号という。受信者は前もって雑音と送信語をどちらも知ることができないので、推定送信語と送信語が異なる場合がある。この場合、復号に失敗したという。復号失敗確率を最小にする復号を最尤復号という。

最尤復号を行うための方法は複数知られており、符号理論では、計算量の小さい復号法を最も優れた復号法とみなす。本稿では、2 項式イデアルのグ

[1] E-mail: daisu-ik@is.aist-nara.ac.jp

レブナ基底を用いた最尤復号法を提案する。本稿で提案した復号法は今まで知られている復号法に比べて計算量が大きいため最も優れた復号法ではないが、代数的な観点から興味深いと考える。

提案最尤復号法のアイデアは、最尤復号を整数計画問題と結び付けた点にある。最尤復号と整数計画問題の関係についてはすでに知られていたが、グレブナ基底を用いた最尤復号は、筆者が知る限り、新規のものである。いくつかの雑音通信路に対して、最尤復号は 2 を法とする制約連立方程式を持つ整数計画問題とみなすことができる。ここで、2 を法としてではなく、整数演算における制約連立方程式を持つ整数計画問題を 2 項式イデアルのグレブナ基底を用いて解く方法は、Conti と Traverso [3] によって 1991 年に提唱されて以来、広く知られるようになった [4,10,11] 。本稿では Conti-Traverso の方法を拡張し 2 を法とする制約連立方程式を持つ整数計画問題を解く、グレブナ基底を用いたアルゴリズムを構成する。

本稿の構成は以下の通り。まず、第 2 節で符号の定義などの準備を行う。次に第 3 節で雑音のある通信路における最尤復号を説明する。第 4 節で [5,6] で提案した 2 項式イデアルのグレブナ基底を用いた最尤復号法の概要を示す。第 5 節で、当研究集会「グレブナー基底の理論的有効性と実践的有効性」で口頭発表した、4 節に現れるアルゴリズムの改良の概略を示す。

2 準備

本稿を通じて、$\mathbf{F}_2 = \{0,1\}$ を 2 元体とする。$n \geq 3$, n : 整数を固定する。n 次元ベクトル空間 $\mathbf{F}_2^n$ とその元 $\mathbf{u} = (u_1, \ldots, u_n), \mathbf{v} = (v_1, \ldots, v_n) \in \mathbf{F}_2^n$ に対して、内積 $\mathbf{u} \cdot \mathbf{v} \in \mathbf{F}_2$ を

$$\mathbf{u} \cdot \mathbf{v} \equiv \sum_{i=1}^{n} u_i v_i \mod 2$$

で定義する。

2.1 2 元線形ブロック符号

この節では、送信者が送信する語の集合 “符号” を定義する。

定義 2.1 (2 元線形ブロック符号). $\mathbf{F}_2$ 上の n 次元ベクトル空間 $\mathbf{F}_2^n$ に対し、部分ベクトル空間 $C \subset \mathbf{F}_2^n$ を **2 元線形ブロック符号 (binary linear block code)** または単に **符号 (code)** という。このとき、n を符号 C の **符号長 (length)** という。

例 2.2. 成分が全て 0 からなる長さ n のベクトルと、成分が全て 1 からなる長さ n のベクトルの集合 $\{(0,0,\ldots,0),(1,1,\ldots,1)\}$ は符号である。この符号を **繰り返し符号 (repetition code)** という。

繰り返し符号の他にも様々な符号が知られている。詳しくは [8] を参照せよ。

符号 C の元を **符号語 (codeword)** という。送信者は符号 C から任意に選んだ元を送信するとする。送信者が選んだ元を **送信語 (transmitted word)** という。受信者は雑音のある通信路を通った長さ n のベクトルを受信する。受信したベクトルの成分は、通信路の定義によって $\mathbf{F}_2$ の元であったり、あるいは実数であったりする。受信者の受け取るベクトルを**受信語 (received word)** という。

符号を定義する上で、次に定義するパリティ検査行列 (parity check matrix) ならびに生成行列 (generator matrix) を考えることがある。

定義 2.3 (パリティ検査行列). 符号 C に対し、$\mathbf{F}_2$ 成分, $d \times n$ 行列 H が

$$C = \{\mathbf{u} | H\mathbf{u} \equiv \mathbf{0} \mod 2, \mathbf{u} \in \mathbf{F}_2^n\}$$

を満たすとき、H を符号 C の**パリティ検査行列**という。

例 2.4 (例 2.2 の続き). 各行に 1 が 2 個現れ、その他は 0 であるような次の $(n-1) \times n$ 行列

$$H = \begin{pmatrix} 1 & 1 & 0 & \dots & & & 0 \\ 0 & 1 & 1 & 0 & \dots & & 0 \\ \vdots & & & & & & \\ 0 & \dots & & & 0 & 1 & 1 \end{pmatrix}$$

は繰り返し符号のパリティ検査行列である。

定義 2.5 (生成行列). 符号 C に対し、C の $\mathbf{F}_2$ 上の基底を行とする、成分 $\mathbf{F}_2$, $\dim C \times n$ 行列を符号 C の**生成行列**という。G を生成行列とするとき、

$$C = \{\mathbf{u}G | \mathbf{u} \in \mathbf{F}_2^{\dim C}\}$$

である。

例 2.6 (例 2.2 の続き). 全ての成分が 1 であるような、次の $1 \times n$ 行列

$$G = (1, 1, \dots, 1)$$

は繰り返し符号の生成行列である。

生成行列 G とパリティ検査行列 H は、

$$GH^T \equiv 0 \mod 2$$

を満たす。ここで、H^T は行列 H の転置を意味する。

2.2 通信路

本節では、雑音のある通信路の数学的モデルを定義する。本稿で提案するグレブナ基底を用いた復号法は、これから定義する対称通信路と加法的白色ガウス雑音通信路に適用できる。本節で定義しないその他の雑音のある通信路の数学的モデルについては、[9] が詳しい。本節を通して、符号長 n および符号 C を固定する。与えられた符号語に対して受信語の決め方により、雑音のある通信路が定義できる。

2.2.1 対称通信路

受信者がバイナリデータを受信すると仮定したとき、雑音のある通信路の中で最も単純で一般的なモデルが対称通信路 (binary symmetric channel) である。

定義 2.7 (対称通信路). $p < 1/2$ を与える。送信語 $\mathbf{c} = (c_1, \ldots, c_n) \in C$ に対し、受信語 $\mathbf{r} = (r_1, \ldots, r_n) \in \mathbf{F}_2^n$ を

$$r_i \equiv c_i + e_i \mod 2$$

$i = 1, \ldots, n$ とする。但し、各 e_i は i に独立に確率 p で 0, 確率 $1-p$ で 1 とする。ベクトル $\mathbf{e} = (e_1, \ldots, e_n) \in \mathbf{F}_2^n$ を対称通信路における**誤りベクトル (error vector)** という。

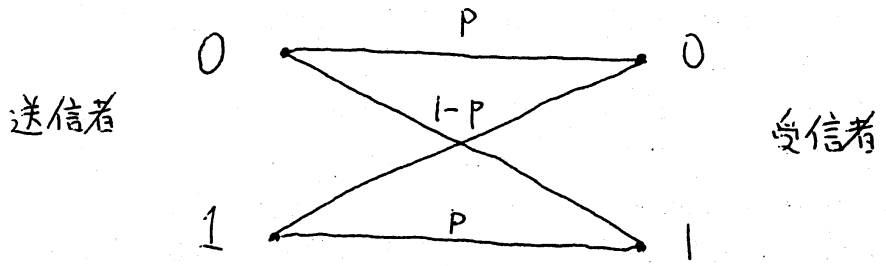

図 2: 対称通信路

2.2.2 加法的白色ガウス雑音通信路

受信者がバイナリデータではなく実数のデータを受信すると仮定したとき、雑音のある通信路の中で最も一般的なモデルが加法的白色ガウス雑音通信路 (AWGN channel, additive white Gaussian noise channel) である。本稿では、2 元位相シフト変調 (BPSK, binary phase shift keying) をシグナル変調に用いた加法的白色ガウス通信路を、単に加法的白色ガウス通信路と呼ぶ。シグナル変調やその他の加法的白色ガウス雑音通信路の定義は [9] を参照せよ。

定義 2.8 (加法的白色ガウス雑音通信路). $\sigma > 0$ を与える。送信語 $\mathbf{c} = (c_1, \ldots, c_n) \in C$ に対し、受信語 $\mathbf{r} = (r_1, \ldots, r_n) \in \mathbf{R}^n$ を

$$r_i = \bar{c}_i + z_i \in \mathbf{R}$$

$i = 1, \ldots, n$ とする。但し、それぞれ $\bar{c}_i = 2c_i - 1 \in \{-1, 1\}$, $z_i \in \mathbf{R}$ は i に独立な平均 0, 分散 σ^2 のガウス分布による乱数とする。ベクトル $\mathbf{z} = (z_1, \ldots, z_n) \in \mathbf{R}^n$ を加法的白色雑音通信路における誤りベクトルという。

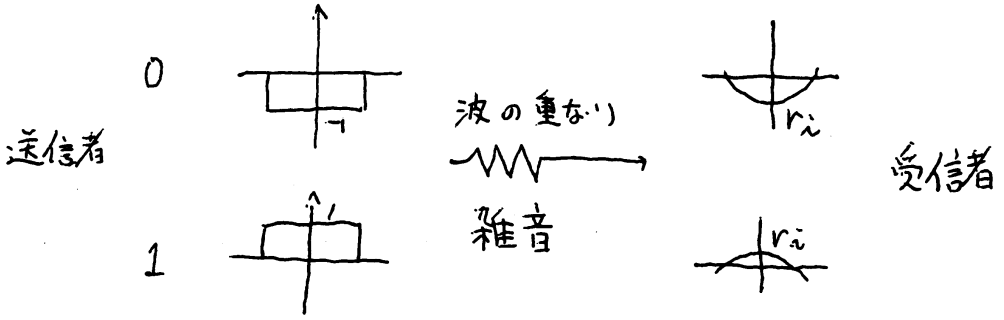

図 3: 加法的ガウス雑音通信路

3 最尤復号

受信者は送信語および誤りベクトルを予め知ることができないと仮定する。このとき、受信者が受信語から送信語を推定することを**復号 (decode)** という。推定する方法あるいはアルゴリズムをそれぞれ**復号法 (decoding)**, **復号アルゴリズム (decoding algorithm)** という。また、推定した符号語を**復号語 (decoded word)** という。復号語が送信語に一致するとき、**復号失敗 (probability of the decoder making a mistake)** という。通信路を固定するとき、全ての復号法において復号失敗確率を最小にする復号法を**最尤復号法 (maximum likelihood decoding)** という。本稿では、任意の符号に対する最尤復号法を対象とする。以下、符号長 n, 符号 C と送信語 $\mathbf{c} \in C$ を固定する。このとき与えられた受信語から復号語の決め方により、復号法あるいは復号アルゴリズムが定義できる。この節を通して $d \times n$ 行列 H を符号 C のパリティ検査行列とする。

3.1 対称通信路における最尤復号

受信語 $\mathbf{r} \equiv \mathbf{c} + \mathbf{e} \mod 2$ に対して、長さ d のベクトル $H\mathbf{r}$ を**シンドローム (syndrome)** という。

$$H\mathbf{r} \equiv H\mathbf{c} + H\mathbf{e} \equiv H\mathbf{r} \mod 2$$

であるから、受信者は誤りベクトル $\mathbf{e}$ を予め知ることができないが、受信語 $\mathbf{r}$ から誤りベクトルの情報 $H\mathbf{e}$ を計算することができることがわかった。

定義 3.1 (シンドローム復号). 長さ n で成分が全て 1 であるベクトルを $\mathbf{t}=(1,\ldots,1)\in\mathbf{F}_2^n$ とおく。受信語 $\mathbf{r}\equiv\mathbf{c}+\mathbf{e} \mod 2$ に対し、

$$\min\{\mathbf{t}\cdot\mathbf{u}|H\mathbf{u}\equiv H\mathbf{r} \mod 2,\mathbf{u}\in\mathbf{F}_2^n\} \tag{1}$$

を満足するベクトルを一つ取り、それを $\tilde{\mathbf{u}}\in\mathbf{F}_2^n$ とする。このとき、復号語を

$$\tilde{\mathbf{c}}\equiv\mathbf{r}+\tilde{\mathbf{u}} \mod 2$$

として決める。このとき、$\tilde{\mathbf{c}}\in C$ である。この復号法を**シンドローム復号 (syndrome decoding)** という。

シンドローム復号は対称通信路における最尤復号法である (詳しくは [8] 参照)。

例 3.2. 符号長 $n=3$, 繰り返し符号 $C=\{(0,0,0),(1,1,1)\}$ に対するシンドローム復号を考える。繰り返し符号 C のパリティ検査行列 H は例 2.4 で見たように

$$H=\begin{pmatrix}1&1&0\\0&1&1\end{pmatrix}$$

である。送信者が符号語 $(1,1,1)\in C$ を対称通信路で送信したとする。また、誤りベクトルが $(0,1,0)\in\mathbf{F}_2^3$ であったとする。このとき、受信者が受信する受信語は

$$(1,0,1)\equiv(1,1,1)+(0,1,0) \mod 2$$

である。このとき、シンドローム $\mathbf{s}\in\mathbf{F}_2^2$ は

$$\mathbf{s}=\begin{pmatrix}1\\1\end{pmatrix}\equiv H\begin{pmatrix}1\\0\\1\end{pmatrix} \mod 2$$

である。線形連立方程式 $H\mathbf{u}\equiv\mathbf{s} \mod 2$ の解は $\{(0,1,0),(1,0,1)\}$ であり、$\mathbf{t}=(1,1,1)$ との内積はそれぞれ $1,2$ である。従って、シンドローム復号による復号語は

$$(1,1,1)\equiv(1,0,1)+(0,1,0) \mod 2$$

であり、復号語は送信語に一致した、つまり復号に成功した。

3.2 加法的白色ガウス雑音通信路における最尤復号

符号語 $\mathbf{c}\in C$ と受信語 $\mathbf{r}=\bar{\mathbf{c}}+\mathbf{z}\in\mathbf{R}^n$ に対して、内積

$$\mathbf{r}\cdot\mathbf{c}=\sum_{i=1}^{n}\{r_i|c_i=1\}$$

を考える。このとき、受信語との内積を最大にする符号語を一つとり、それを復号語とする復号法は最尤復号法であることが知られている (詳しくは [7] 参照)。この復号法をパリティ検査行列 H を用いて書き直すと以下のようになる。

定義 3.3 (加法的白色ガウス雑音通信路における最尤復号). 受信語 $\mathbf{r} = \bar{\mathbf{c}} + \mathbf{z} \in \mathbf{R}^n$ に対し、

$$\max\{\mathbf{r} \cdot \mathbf{u} | H\mathbf{u} \equiv \mathbf{0} \mod 2, \mathbf{u} \in \mathbf{F}_2^n\} \tag{2}$$

を満足するベクトルを一つとり、それを $\tilde{\mathbf{u}} \in \mathbf{F}_2^n$ とする。$\tilde{\mathbf{u}} \in C$ である。このとき、復号語を $\tilde{\mathbf{c}}$ とする。

対称通信路における最尤復号法を**硬判定最尤復号法**、加法的白色ガウス雑音通信路における最尤復号法を**軟判定最尤復号法**という。

例 3.4. 符号長 $n = 3$, 繰り返し符号 $C = \{(0,0,0),(1,1,1)\}$ に対する軟判定最尤復号を考える。送信語を $\mathbf{c} = (0,0,0)$ とする。このとき、$\bar{\mathbf{c}}$ はその定義から $\bar{\mathbf{c}} = (-1,-1,-1)$ となる。誤りベクトルを $\mathbf{z} = (2,0,-1) \in \mathbf{R}^3$ とする。このとき受信語は $(1,0,-2)$ となる。受信語と符号語 $(0,0,0),(1,1,1)$ の内積はそれぞれ $0,-1$ であるので、復号語を $(0,0,0)$ とする。このとき、復号は成功した。

3.3 最尤復号と計算量

硬判定最尤復号および軟判定最尤復号は、それぞれ式 (1) および式 (2) を満足する $\tilde{\mathbf{u}}$ を求めることで行うことができることがわかった。ここで、集合 $\{\mathbf{u}|H\mathbf{u} \equiv H\mathbf{r} \mod 2, \mathbf{u} \in \mathbf{F}_2^n\}$, $\{\mathbf{u}|H\mathbf{u} \equiv \mathbf{0} \mod 2, \mathbf{u} \in \mathbf{F}_2^n\}$ はどちらも有限集合であるから、ベクトル $\mathbf{t}$ との内積の最小値および $\mathbf{r}$ との内積の最大値は、それぞれの集合の元との内積を全て計算することで有限時間で計算することができる。

しかし、工学的な要求から「できるだけ少ない計算回数」で最尤復号法を行うことが求められる。

そこで、本稿では硬判定最尤復号においては $\mathbf{F}_2$ の元の四則演算の回数を、軟判定最尤復号においては、実数の四則演算および大小比較の回数[2]を数え上げ、その回数が少ないほど「よい復号法」であると決める。

一般に、復号法を提案したときに計算回数を決定するのは困難であることが多い。そこで、符号 C と通信路の雑音の確率 (硬判定ならば確率 p, 軟判定ならば分散 σ^2) を固定し、最大計算回数および平均計算回数を計算機模擬により評価することがしばしば行われる。最大計算回数や平均計算回数を決定するのも困難なことが多いので、復号の回数を指定して (たとえば 1000 回

[2]軟判定最尤復号法を提案する論文では、$\mathbf{F}_2$ の元の計算回数は復号法の性能比較においてしばしば無視される。

の復号試験を行い)、そのときの最大および平均を求めることもしばしば行われる。

4 2項式イデアルのグレブナ基底を用いた最尤復号

本節では、[5,6] で提案した 2 項式イデアルのグレブナ基底を用いた最尤復号を説明する。本稿を通して、k を任意の有限体とする。また、符号長 n および符号 C を固定し、符号のパリティ検査行列を H, 生成行列を G とする。

4.1 最尤復号の標準形

定義 3.1 の中に現れた式 (1) と定義 3.3 の中に現れた式 (2) を統一して扱うために、次のような問題を考える。

問題 4.1 (最尤復号の標準形). 成分 $\mathbf{F}_2$, $d \times n$ 行列 $A = (a_{i,j})$, 成分 $\mathbf{F}_2$, 長さ d のベクトル $\mathbf{b} = (b_i)$ および成分 $\mathbf{R}$, 長さ n のベクトル $\mathbf{w} = (w_j)$ を与える $(i = 1, \ldots, d, j = 1, \ldots, n)$。このとき、

$$\min\{\mathbf{w} \cdot \mathbf{u} | A\mathbf{u} \equiv \mathbf{b} \mod 2, \mathbf{u} \in \mathbf{F}_2^n\} \tag{3}$$

を満足する $\tilde{\mathbf{u}} \in \mathbf{F}_2^n$ を求めよ。

このときそれぞれ、硬判定最尤復号は $A = H, \mathbf{b} = H\mathbf{r}, \mathbf{w} = \mathbf{t}$, 軟判定最尤復号は $A = H, \mathbf{b} = \mathbf{0}, \mathbf{w} = -\mathbf{r}$ とすれば標準形に一致することがわかる。

4.2 w が非負のとき

説明のために、まず w が非負実数成分のときに問題 4.1 を解くアルゴリズムを与え、その後で w が任意の実数成分の場合に拡張する。

行列 A の各列ベクトルを $\mathbf{a}_j$ とする $(j = 1, \ldots, n)$。k 係数 $(n+d)$ 変数多項式環 $R = k[X_1, \ldots, X_n, Y_1, \ldots, Y_d]$ のイデアル I_A を次のように定義する。

$$I_A = \langle X_1 - \mathbf{Y}^{\mathbf{a}_1}, \ldots, X_n - \mathbf{Y}^{\mathbf{a}_n}, Y_1^2 - 1, \ldots, Y_d^2 - 1 \rangle$$

また、$k[X_1, \ldots, X_n]$ の項順序 $\prec_{\mathbf{X}}$ および $k[Y_1, \ldots, Y_d]$ の項順序 $\prec_{\mathbf{Y}}$ を任意にとり、項順序 $\prec_{\mathbf{X},\mathbf{w},\mathbf{Y}}$ を次のように決める。

定義 4.2 (適応的項順序 (adapted monomial order)). $\mathbf{X}^{\mathbf{u}}\mathbf{Y}^{\mathbf{s}} \prec_{\mathbf{X},\mathbf{w},\mathbf{Y}} \mathbf{X}^{\mathbf{v}}\mathbf{Y}^{\mathbf{t}} \Leftrightarrow$

1. $\mathbf{Y}^{\mathbf{s}} \prec_{\mathbf{Y}} \mathbf{Y}^{\mathbf{t}}$
2. $\mathbf{s} = \mathbf{t}$ かつ $\mathbf{w} \cdot \mathbf{u} < \mathbf{w} \cdot \mathbf{v}$

3. $\mathbf{s}=\mathbf{t}$ かつ $\mathbf{w}\cdot\mathbf{u}=\mathbf{w}\cdot\mathbf{v}$ かつ $\mathbf{X}^{\mathbf{u}} \prec_{\mathbf{X}} \mathbf{X}^{\mathbf{v}}$

このとき、イデアル I_A の適応的項順序 $\prec_{\mathbf{X},\mathbf{w},\mathbf{Y}}$ に対するグレブナ基底 $\mathcal{G}$ を用いて問題 4.1 を解くアルゴリズムを構成できる。

Algorithm 1 標準形を解くアルゴリズム

Input: $A, \mathbf{b}, \mathbf{w}$. 但し $\mathbf{w}$ は非負実数ベクトル。

Output: $A, \mathbf{b}$ および非負実数ベクトル $\mathbf{w}$ に対する問題 4.1 の解 $\mathbf{u}\in\mathbf{F}_2^n$

1: イデアル I_A の適応的項順序 $\prec_{\mathbf{X},\mathbf{w},\mathbf{Y}}$ に対するグレブナ基底 $\mathcal{G}$ を計算する。

2: 単項式 $\mathbf{Y}^{\mathbf{b}}$ の $\mathcal{G}$ による標準形を求め、標準形の $\mathbf{X}$ のベキを出力する。

ここで、単項式 $\mathbf{Y}^{\mathbf{b}}$ の $\mathcal{G}$ による標準形は $\mathbf{X}$ の単項式になることが保証されるのでそのベキは一意的に定まる。

4.3 wが負の成分を含むとき

ベクトル $\mathbf{w}$ が負の成分を含むときは、適応的項順序を定義することができない (2 番目の条件が項順序の定義に反するので)。そこで次のようなことを考える。

定義 4.3 (Lawrence Lifting). $d\times n$ 行列 A に対し、$(d+n)\times 2n$ 行列

$$\Lambda(A)=\begin{pmatrix} A & \mathbf{0} \\ \mathbf{1} & \mathbf{1} \end{pmatrix}$$

を行列 A の Lawrence Lifting という。但し、$\mathbf{0}$ は $d\times n$ 零行列、$\mathbf{1}$ は $n\times n$ 単位行列とする。

さて、$\mu=\max\{w_i | i=1,\ldots,n\}\in\mathbf{R}$ とおき、$\mathbf{w}'=(w'_1,\ldots,w'_{2n})\in\mathbf{R}^{2n}$ を

$$w'_i=\begin{cases} \mu-w_i & (i=1,\ldots,n) \\ \mu & (i=n+1,\ldots,2n) \end{cases}$$

として与える。このとき、w'_i はすべての $i=1,\ldots,2n$ において非負である。また、$\mathbf{b}'=(b'_1,\ldots,b_{d+n})\in\mathbf{F}_2^{d+n}$ を

$$b'_i=\begin{cases} b_i & (i=1,\ldots,d) \\ 1 & (i=d+1,\ldots,d+n) \end{cases}$$

として与える。$\mathbf{w}'$ は非負実数ベクトルなので、前節におけるアルゴリズム 1 を用いて $\Lambda(A),\mathbf{b}',\mathbf{w}'$ に対する標準形の解を求めることができる。このとき、$\Lambda(A),\mathbf{b}',\mathbf{w}'$ に対する標準形の解を $\tilde{\mathbf{u}}'=(u'_1,\ldots,u'_{2n})$ とすると、$A,\mathbf{b},\mathbf{w}$ の標準形の解は $\tilde{\mathbf{u}}'$ の左半分、すなわち $(u'_1,\ldots,u'_n)$ であることが従う。

5 アルゴリズムの改良

前 4 節で、標準形を解くアルゴリズム 1 を紹介した。しかし、このアルゴリズムはグレブナ基底の計算を行う最初のステップで時間がかかるために、他の最尤復号アルゴリズムと比べて計算回数が著しく大きい。そこで、アルゴリズムを改善することを考える。

$d \times n$ 行列 A, ベクトル $\mathbf{b}, \mathbf{w}$ は前 4 節と同じとする。但し、$d < n$ とし、A は full rank であると仮定する。

行列 A に対して、成分 $\mathbf{F}_2$, $(n-d) \times n$ 行列 $A^\star$ が

$$A^\star A^T \equiv 0 \mod 2$$

を満たすとする。但し、A^T は行列 A の転置行列を意味する。このような $A^\star$ は必ず存在する。$A^\star$ の各行ベクトルを $\mathbf{a}_i^\star$ とする $(i = 1, \ldots, n-d)$。

k 係数 n 変数多項式環 $k[X_1, \ldots, X_n]$ のイデアル $J_{A^\star}$ を次のように定義する。

$$J_{A^\star} = \langle \mathbf{X}^{\mathbf{a}_1^\star} - 1, \ldots, \mathbf{X}^{\mathbf{a}_{n-d}^\star} - 1, X_1^2 - 1, \ldots, X_n^2 - 1 \rangle$$

次に、$k[X_1, \ldots, X_n]$ の項順序 $\prec_{\mathbf{X}}$ を任意にとり、同じく $k[X_1, \ldots, X_n]$ の項順序 $\prec_{\mathbf{X},\mathbf{w}}$ を次のようにして与える。

定義 5.1 (重みつき項順序 (weighted monomial order)). $\mathbf{X}^{\mathbf{u}} \prec_{\mathbf{X},\mathbf{w}} \mathbf{X}^{\mathbf{v}}$ $\Leftrightarrow$

1. $\mathbf{w} \cdot \mathbf{u} < \mathbf{w} \cdot \mathbf{v}$
2. $\mathbf{w} \cdot \mathbf{u} = \mathbf{w} \cdot \mathbf{v}$ かつ $\mathbf{X}^{\mathbf{u}} \prec_{\mathbf{X}} \mathbf{X}^{\mathbf{v}}$

このとき、イデアル $J_{A^\star}$ の重みつき項順序 $\prec_{\mathbf{X},\mathbf{w}}$ に対するグレブナ基底を用いて問題 4.1 を解くアルゴリズムを構成できる。

Algorithm 2 標準形を解く改良アルゴリズム

Input: $A^\star, \mathbf{b}, \mathbf{w}$, および $A\mathbf{u} \equiv \mathbf{b} \mod 2$ の解 $\mathbf{u} \in \mathbf{F}_2^n$。但し $\mathbf{w}$ は非負実数ベクトル。

Output: $A, \mathbf{b}$ および非負実数ベクトル $\mathbf{w}$ に対する問題 4.1 の解 $\mathbf{u} \in \mathbf{F}_2^n$

1: イデアル $J_{A^\star}$ の重みつき項順序 $\prec_{\mathbf{X},\mathbf{w}}$ に対するグレブナ基底 $\mathcal{G}$ を計算する。

2: 単項式 $\mathbf{X}^{\mathbf{u}}$ の $\mathcal{G}$ による標準形を求め、標準形の $\mathbf{X}$ のベキを出力する。

グレブナ基底の計算は、変数と生成元の個数が少なければ少ないほど計算が速いことが経験的に知られている (詳しくは [1] 参照)。したがって、本節で紹介した改良アルゴリズムのほうが計算回数が少ないのではないかと推定

注意 5.2. 硬判定最尤復号においては、$A = H, \mathbf{b} = H\mathbf{r}$ であるので、アルゴリズム 2 において $\mathbf{u} = \mathbf{r}$ とすれば適用できる。軟判定最尤復号については、$A = \Lambda(H), \mathbf{b} = (0, \ldots, 0, 1, \ldots, 1)$ である。そこで、符号語 $\mathbf{c} = (c_1, \ldots, c_n) \in C$ を任意に選び、$\mathbf{F}_2^n \ni \hat{\mathbf{c}} \equiv (1, \ldots, 1) - \mathbf{c}$ とする。長さ $2n$ のベクトル $\mathbf{u}$ を $(\mathbf{c}, \hat{\mathbf{c}})$ とすれば、アルゴリズム 2 が適用できる。

注意 5.3. $\mathbf{w}$ に負の成分が含まれる場合、前 4 節と同様に、行列 A の Lawrence Lifting $\Lambda(A)$ を考える。このとき、

$$(A^*A^*)\Lambda(A)^T \equiv 0 \mod 2$$

であるから、$(n-d) \times 2n$ 行列 $(\Lambda(A))^*$ を

$$(\Lambda(A))^* = (A^*A^*)$$

として取ることができる。

参考文献

[1] Thomas Becker, Volker Weispfenning, and Heinz Kradel. *Gröbner bases – A Computational Approach to Commutative Algebra.* Springer-Verlag, 1993.

[2] Arjeh M. Cohen, Hans Cuypers, and Hans Sterk, editors. *Some Tapas of Computer Algebra*, volume 4. Springer, 1998.

[3] Pasqualina Conti and Carlo Traverso. Buchberger algorithm and integer programming. In Harold F. Mattson, Teo Mora, and T. R. N. Rao, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (AAECC-9)*, number 539 in LNCS, pages 130–139. Springer-Verlag, October 1991.

[4] David Cox, John Little, and Donal O'Shea. *Using Algebraic Geometry.* Springer-Verlag, 1998.

[5] Daisuke Ikegami and Yuichi Kaji. A soft-decision MLD algorithm for linear block codes using Gröbner bases. In 第 *24* 回情報理論とその応用シンポジウム *(SITA 2001)*, pages 545–548, December 2001.

[6] Daisuke Ikegami and Yuichi Kaji. Maximum likelihood decoding for linear block codes using Gröbner bases. submitted to IEICE Transaction on Fundamentals, 8 2002.

[7] Shu Lin, Tadao Kasami, Toru Fujiwara, and Marc Fossorier. *Trellises and Trellis-Based Decoding Algorithms for Linear Block Codes.* Kluwer Academic Publishers, 1998.

[8] F. Jessie MacWilliams and Neil J. A. Sloane. *The Theory of Error-Correcting Codes*, volume 16 of *North-Holland Mathematical Library*. North-Holland, ninth edition, 1996.

[9] John G. Proakis. *Digital Communications*. McGraw-Hill, Inc., second edition, 1989.

[10] Bernd Sturmfels. *Gröbner Bases and Convex Polytopes*, volume 8 of *University Lecture*. American Mathematical Society, 1995.

[11] Günter M. Ziegler. *Gröbner Bases and Integer Programming*, chapter 7, pages 168–183. Volume 4 of Cohen et al. [2], 1998.